Parameter-Efficient Fine-Tuning of RoBERTa for Text Classification

Nick Ferrara

Deep Learning
Professor Chinmay Hegde
github.com/NxFerrara/lora-finetuning

Abstract

This project explores parameter-efficient fine-tuning of RoBERTa for text classification using Low-Rank Adaptation (LoRA). We compare two approaches: a baseline configuration with LoRA applied only to query matrices, and an improved configuration targeting query, key, and value matrices with enhanced training parameters. On the AG News dataset, our improved approach achieved significantly better performance while maintaining under 1 million trainable parameters. Our findings highlight how even small modifications to fine-tuning strategy can dramatically improve performance without exceeding parameter constraints.

Introduction

Natural language processing tasks often require fine-tuning large pre-trained language models, which can be computationally expensive and memory-intensive. Parameter-Efficient Fine-Tuning (PEFT) techniques address this by updating only a small subset of a model's parameters while keeping most of the pre-trained weights frozen.

Low-Rank Adaptation (LoRA) is a PEFT technique that inserts trainable matrices into the layers of a transformer model. This reduces the number of trainable parameters while maintaining performance.

In this project, we applied LoRA to fine-tune RoBERTa for text classification on the AG News dataset, which contains news articles categorized into four classes: World, Sports, Business, and Sci/Tech. We compared two LoRA configurations and training approaches.

Methodology

Base Model and Dataset

We used the RoBERTa model as our foundation, which contains approximately 125 million parameters. For our classification task, we utilized the AG News dataset, which consists of 120,000 training examples evenly distributed across four categories.

The dataset was preprocessed using the RoBERTa tokenizer and split into training and validation sets to monitor performance during training.

LoRA Configurations

We experimented with two different LoRA configurations:

- **Baseline Configuration:** Applied LoRA only to the query matrices in the attention mechanism, with rank r=2, alpha=4, and dropout=0.05.
- **Improved Configuration:** Extended LoRA to query, key, and value matrices with rank r=4, alpha=8, and dropout=0.1.

Table 1 summarizes the key differences between these configurations:

Parameter	Baseline	Improved
Target Modules	Query only	Query, Key, Value
Rank (r)	2	4
Alpha	4	8
Dropout	0.05	0.1
Trainable Parameters	630,532	814,852
% of Total Parameters	0.5033%	0.6495%

Table 1: Comparison of LoRA configurations used in our experiments.

Training Approach

We also implemented significant improvements in our training methodology between the baseline and improved configurations:

Training Parameter	Baseline	Improved
Optimizer	SGD	AdamW
Learning Rate	5e-6	2e-4
Weight Decay	None	0.01
LR Scheduler	None	Linear with warmup
Warmup Steps	None	100

Table 2: Comparison of training parameters between baseline and improved configurations.

Both models were trained for a maximum of 600 steps on the AG News dataset using identical hardware. We used a batch size of 8 for training and 64 for evaluation.

Results

Model Performance

Our baseline and improved configurations showed significant differences in performance:

Metric	Baseline	Improved
Final Validation Accuracy	22.66%	90.94%
Final Training Loss	1.395	0.267

Table 3: Performance comparison between baseline and improved models.

The baseline model's accuracy remained stagnant at 22.66%, even slightly worse than random guessing (25% for 4 classes). This indicates that the model was unable to effectively learn the classification task. This is likely due to the small (and constant) learning rate.

In contrast, the improved model showed continuous improvement throughout training, achieving a validation accuracy of 90.94%, demonstrating effectiveness in learning with our configurations.

Analysis of Improvements

We attribute the significant performance improvement to three key factors:

- **Expanded LoRA Targets:** Targeting all attention matrices (query, key, value) provided more capacity for the model to learn task-specific adaptations.
- **Optimizer Selection:** Switching from SGD to AdamW dramatically improved optimization for the transformer architecture.
- **Learning Rate and Scheduling:** The 40× increase in learning rate combined with proper scheduling allowed the model to escape local minima and converge to a better solution.

This substantial improvement came with only a small increase in trainable parameters, from 630,532 (0.5033% of total) to 814,852 (0.6495% of total).

Conclusion

Our experiment demonstrates the importance of configuration choices in parameter-efficient fine-tuning. While LoRA inherently reduces the number of trainable parameters, the specific configuration of target modules and training parameters significantly impacts performance.

Our improved configuration achieved a 90.94% accuracy while fine-tuning only 0.6495% of the model's parameters, demonstrating that application of PEFT techniques can yield excellent results with minimal computational overhead.

Future work could explore with larger language models and more complex tasks, as well as experimenting with other PEFT techniques.

References

[1] Zhang, A.; Lipton, Z. C.; Li, M.; and Smola, A. J. 2023. Dive into Deep Learning, 1st Edition. Cambridge University Press.

[2] Beji, V. 2025. Deep Learning Spring 2025: Project 2. https://www.kaggle.com/competitions/deep-learning-spring-2025-project-2/overview. Kaggle.

[3] OpenAI. 2025. ChatGPT. https://chat.openai.com/. Used for assistance in outlining report structure.